REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE
3. REPORT TYPE AND DATES COVERED	FINAL/15 Jun 94 - 15 Dec 94
4. TITLE AND SUBTITLE	USER-FRIENDLY MICROCOMPUTER INTERFACE WITH OPTIMIZATION LANGUAGES AF94-007
5. FUNDING NUMBERS	F49620-94-C-0045 61102F 3005/SS
6. AUTHOR(S)	MR SAM SAVAGE
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	PRIMAL SOLUTIONS 124 UNIVERSITY AVENUE, STE 300 PALO ALTO, CA 94301
8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	AFOSR/NM 110 DUNCAN AVE, SUTE B115 BOLLING AFB DC 20332-0001
10. SPONSORING/MONITORING AGENCY REPORT NUMBER	F49620-94-C-0045
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION / AVAILABILITY STATEMENT	APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

AFOSR-TR- 95-0225

13. ABSTRACT (Maximum 200 words)

All Phase I goals were met. Add-in command for IMPROV were developed that allowed specification of variables and constraints, translation, optimization, and display of results.

19950403 136

14. SUBJECT TERMS	
15. NUMBER OF PAGES	
16. PRICE CODE	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR(SAME AS REPORT)

User-friendly Microcomputer Interface with Optimization Languages AF94-007

Contract Code FA9550

Final Technical Report
30 September, 1994

Accesion For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

I. Objectives

What They Were

The Objectives of Phase I were to:

1) Develop software to translate valid IMPROV models into files which can be loaded into LINGO and optimized, whereupon the answers are returned to IMPROV. This is the heart of the system.
2) Strive to keep the translator as general as possible so that other user interfaces and modeling languages may be supported with minimal redesign.
3) Develop add-in commands to the IMPROV user-interface which allow for:
    a) Specification of the variables, objective, and constraints of the model
    b) Translation to LINGO and optimization
    c) Displaying optimal answers and dual values returned from LINGO
4) Provide Alpha test programs to mathematical modelers in military and industrial settings to obtain feedback on design and capabilities.

What We Did

We not only accomplished all of the above, we actually completed and shipped a commercial IMPROV based optimization package using Lindo Systems LINGO optimization language (two copies enclosed). This product has exceeded our development expectations for this phase of the contract.

Questions and Answers

The following questions, raised in our proposal have been answered as shown.

Question:

1) Can completely arbitrary IMPROV models be translated by the Phase I software or must the user impose some additional structure on the model?

Answer:

Although examples probably exist, we came across a sensible IMPROV model which was not properly translated.

Question:

2) If arbitrary IMPROV models cannot be translated by the Phase I translator, how much additional work would be required to accomplish this?

Answer: NA

Question:

3) Does IMPROV provide a sufficiently rich syntax to "span" the models which may be expressed by the target modeling languages?

Answer:

IMPROV does provides a rich enough syntax to span most optimization problems. However, like the standard spreadsheet it is deficient in indirect referencing of individual elements of sets. For example it is difficult to describe the set {X(I,J) for I<J}. Although this is not important in most models, it represents an area for improvement.

Question:

4) How much work is required to translate to GAMS, AMPL and MPL?

Answer:

It is felt that the translation code is so general that only a matter of weeks would be required per additional modeling language.

Question:

5) How much work is required to translate from other user interfaces such as COMPETE, TM/1 (a multi-dimensional database), or other databases such as dBASE or PARADOX?

Answer:

A number of products other than IMPROV now store and display multi-dimensional data. This category has been called On Line Analytical Processing. or OLAP (see interim report of Sept. 14th, attached). All of these products allow the user to view multi-dimensional data from many perspectives, but their ability to do mathematical modeling varies. We believe that the modeling ability of the environment itself will be the limiting factor in adding optimization, rather than the modification of our translator.

Question:

6) Overall, is IMPROV paradigm an improvement over current modeling technology?

Answer:

IMPROV has proven to be much more powerful than the standard spreadsheet and much easier to use than the algebraic modeling languages. We have seen several software reviews using the word "Improv-like" to describe new products. We believe that the multi-dimensional paradigm is here to stay, and that it is a great improvement over any current general system for mathematical optimization modeling.

II. Status

A. Optimization Product Shipped

Primal Solutions has now developed commercial software to integrate Optimization, Multivariate Regression, and Time Series Analysis with Lotus IMPROV.

B. Lotus Cancels Future Development of Improv

The same week that hundreds of demo disks were being mailed out to potential customers requesting information, Lotus announced that it had canceled development on future versions of IMPROV thereby immediately poisoning the market.

After talking to people at Lotus, we know there is still interest in Improv within the company, and that Lotus is now looking for other potential firms to sell the product to. Furthermore, Lotus Management has stated to Primal Solutions that in presenting Improv to such potential firms, they are presenting Primal Solutions Analytics as a means of adding value to the Improv product. In the event that a large firm does buy Improv, a significant market could be developed for the product we have already developed. However, there is no guarantee that this will happen, and we must be prepared for the contingency that Improv will no longer be a viable product.

C. On Line Analytical Processing (OLAP)

Since Lotus canceled its plans for the development of future versions of Improv, we have broadened our focus to include other modeling and data analysis environments.

As it turns out, Improv was just one manifestation of a broad trend in multi-dimensional analysis which is known as "On Line Analytical Processing" as discussed in the interim report of September 14.

Primal Solutions is currently seeking other OLAP environments with which to integrate their technology. The current products are well designed as serve as an excellent prototype for future work. Lotus Corp.

III. Interactions

The following interactions have arisen in conjunction with this contract.

- On Friday, June 17, Dr. Sam Savage visited AMC Studies and Analysis Flight, Air Mobility Command Scott AFB to install the prototype Improv/Lingo optimization system as developed up to that point. He met with several members of the Analysis team, and provided initial training in the system to Alan Whisman and Capt. Bruce Schinelli. The system was installed on Capt. Schinelli's computer. Lt. Mark Grabau of tanker command also expressed interest in installing the system and was granted permission.

- During that visit Dr. Savage also demonstrated with 1-2-3 and What'sBest!, a prototype of a stochastic optimization model for routing supply aircraft in the face of uncertain demand as described in the interim report of August 5th.

- On Wednesday July 27th, Dr. Savage gave a presentation at the Naval Postgraduate School in Monterey entitled "Object Oriented Optimization". The abstract follows:

  Mathematical optimization has traditionally required a lengthy mathematical modeling phase followed by a short computer run. The introduction of modeling languages (GAMS, AMPL, LINGO etc.) has led to what Rick Rosenthal has called "Rapid Deployment Optimization". The process of modeling may be further accelerated by assembling optimization models from libraries of commonly used modules (optimization objects). A Prototype system based on Lotus IMPROV will be demonstrated.

- On Monday August 15th, Dr. Savage gave a presentation at the 15th International Symposium on Mathematical Programming, Ann Arbor, entitled "Lotus Improv as a Mathematical Modeling Language". The abstract follows:

  The electronic spreadsheet through sheer numbers has become an important mathematical modeling language. However, this environment has several serious drawbacks:
  1) Spreadsheet models are not independent of their Data.
  2) Spreadsheet models are fundamentally at most 3 dimensional.
  3) Spreadsheet models are difficult to document and verify.
  Improv, a new modeling tool from Lotus Development, appears to resolve these and other problems.

- Dr. Savage has written a paper entitled "The DNA of Decision Science" to appear in a volume <u>on Education in a Research University</u> edited by Kenneth Arrow, B. Curtis Eaves, and Ingram Olkin. See Appendix for 1st draft.

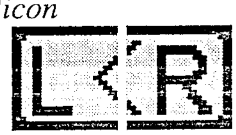
```
@FOR(PLANTS(J) : SUPPLIED(J) < CAPACITY(J));
@FOR(WAREHOUSE(I) : RECEIVED(I) > DEMAND(I));
MIN = @SUM( ROUTES(I,J): VOLUME(I,J) * COST(I, J));
@FOR(WAREHOUSE(I) : RECEIVED(I) = @SUM( PLANTS(J): VOLUME(J, I)));
@FOR(PLANTS(J) : SUPPLIED(J) = @SUM( WAREHOUSE(I): VOLUME(J, I)));
DATA:
CAPACITY = 310, 260, 280;
COST = 10, 8, 6, 5, 4, 6, 5, 4, 3, 6, 3, 4, 5, 5, 9;
DEMAND = 180, 80, 200, 160, 220;
ENDDATA

END
```

B. Bicons

A Challenge has been to develop an Icon based approach to constraint specification. In the past, Icons have been used for unary (single argument) operations. Constraint specification has a left hand and right hand side, and is therefore binary. We have developed a system using two adjacent Icons (the Bicon) to solve this problem as shown below .

The Constraint Bicon



Syntax

Select left hand side  Select right hand side 

V. Summary

The work done under this contract is a continuation of an effort initiated by Dr. Savage in 1985 to facilitate mathematical optimization modeling in widely adopted user friendly environments. We believe the results represent one of the most powerful tools available today for developing general mathematical optimization models.

As such, there is great disappointment it Lotus' decision to cancel future IMPROV development. However, the current product serves as both a successful proof of concept, and as a source of software technology with which to move forward. We are grateful to the Air Force Office of Scientific Research for their support.

The DNA of Decision Science

Sam L. Savage

This work supported in part by the Air Force Office of Scientific Research, Bolling AFB

The decision sciences are evolving rapidly. For a curriculum of study in this area to survive, it must evolve in parallel. By considering a trait common to many evolving systems, I will explore a potential path of future evolution for the decision sciences and their teaching.

The Origins of Life etc.

Evolutionary Aggregation

Evolving systems are often comprised of building blocks which were once at the evolutionary forefront of the system themselves.

In biology, for example, ammonia and methane were at one time, the "King of Beasts". Later, they formed the basis of amino acids which in turn became the components of DNA.

This may be viewed as an evolutionary process of *aggregation* of the simple into the complex. In recent discussions with B. Curtis Eaves, he has suggested the term *sublimation* to describe what happens to the components which comprise the aggregate. I suspect there may be a more formal statement of this principle of aggregation in the emerging theory of Complexity. For example, see Waldrop (1992).

Memes

In *The Selfish Gene,* Dawkins (1976) defines a cultural analog of the gene as the "meme". Memes are the building blocks of ideas and can mutate, evolve or become extinct as they are passed from generation to generation. Evolutionary Aggregation is often discernible in the memes associated with technologies.

For example, the ancient memes of mechanical engineering; the wheel, lever and pump, formed the basis for the heat engines of the early 1800's. For nearly half a century, these "engines" were only operated by "engineers", in large industrial applications such as the pumping of flooded mine shafts. Eventually, as sublimated along with the transmission and pneumatic tire into the aggregate of the automobile, the engine was finally of benefit to the individual.

The memes of electronics; resistors, capacitors, coils and vacuum tubes, were the building blocks of early computers in the 1940's. The transistor led to the aggregation of far more electronic components into the computers of the 1960's. Until the early 1980's, computers were only operated by engineers. Integrated circuitry has raised the level of aggregation to millions of components per square inch, and led to computers for individual use.

For technological systems, the aggregation principle may be restated as:

Today's systems are often tomorrow's subsystems.

Evolution and Ergonomics

Much of modern industrial design is rightly focused on the *ergonomics*, or ease of use of technologies. The cases of the automobile and personal computer show that:

Complex systems are often easier to use than their simpler ancestors

Today's automobiles are far more complex than those of Henry Ford's, yet far easier to drive. The starting crank attached directly to the engine, which could occasionally break your wrist, has been replaced by the ignition key.

The computer sitting on my lap, is more complex than John Von Neumann could have dreamt of stuffing into a dirigible hanger. Yet it is much easier to use than his first simple machine. I wonder if it would have been possible in 1947 to convince Von Neumann that someday consumers would be plugging in their own tiny 8 megabyte memory cards. Arthur C. Clarke's version of this is:

Any Sufficiently Advanced Technology is Indistinguishable from Magic

Standardization - The Network Externality

Another important factor in technological evolution is the establishment of standards such as the distance between railroad tracks, or the number of threads per unit length on bolts. These are known in economics as *network externalities*, and can spell the difference between the proliferation and extinction in technologies.

As a classic example of a network externality, imagine choosing between two telephones. The first, provides call forwarding and a built in answering machine, but was manufactured in France. and is incompatible with the U.S. phone network. The second, is an old rotary dial type manufactured to U.S. standards.

Some technological standards evolve by design, others by chance. But whatever their origin, once established they must not be ignored.

Decision Science

How different is decision science from mechanics or electronics?

Its first memes were the laws of arithmetic required for decisions involving the sharing and bartering of commodities. Technologies built on arithmetic, such as algebra and probability, form the building blocks of modern decision science. Because most decision makers in industry and government do not view their decisions in these algebraic and probabilistic terms, decision science was, until recently, in a state similar the heat engine in nineteenth century: applicable only in large industrial applications, but of no use at the individual level. I refer to this as an "Algebraic Curtain" separating the Decision Maker from Decision Science.

THE DEVELOPMENTAL NECESSITIES OF APPLICATIONS - DNA

The meteoric evolution of the microcomputer has resulted in a new level of aggregation for the Decision Sciences. The algebraic and probabilistic building blocks are being sublimated into computerized models containing what I refer to as the Developmental Necessities of Applications, or DNA. Unlike the algebraic representations of the past, this digital DNA is "alive". It contains just enough representative data so that a user can experimentally infer its structure. It may be expanded, and fed data, and it yields results as output, with little mathematical understanding on the part of the user.

Transformation vs. Formulation

This differs from traditional decision science, in that the decision maker does not formulate a model from scratch, merely transforms an existing one. As an analogy, consider the daunting task of constructing a flamingo from scratch, from amino acids as opposed to the trivial task of incubating a fertilized flamingo egg.

Two Examples

I will present two examples of widely disseminated decision science DNA. One requiring no transformation whatever, the other, simple scaling and data entry.

<u>1. Optical Character Recognition</u>
Optical Character Recognition (OCR) is one of the few examples of decision science truly accessible by the masses. Its DNA is either bound with commercially available software to be used with scanners, or may be hard wired within the ROM of electronic devices upon which one writes with a stylus. Although these programs are not yet error free, they are continually improving in accuracy. The neural nets or other complex artificial intelligence upon which they are based, are so thoroughly sublimated as to be invisible. No transformation of the DNA is required in its application.

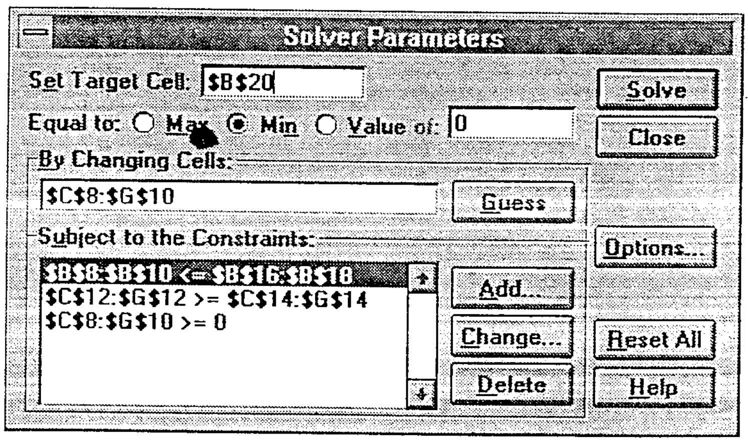
<u>2. Spreadsheet DNA for a Linear Program</u>
For several years, Microsoft has included limited mathematical optimization capability with its Excel spreadsheet of which on the order of 10^6 are shipped annually. The package includes small worksheet examples for several applications. Below is the dense transportation linear programming model.

	A	B	C	D	E	F	G	H
1	Example 2: Transportation Problem.							
2	Minimize the costs of shipping goods from production plants to warehouses near metropolitan demand							
3	centers, while not exceeding the supply available from each plant and meeting the demand from each							
4	metropolitan area.							
6			*Number to ship from plant x to warehouse y (at intersection):*					
7	*Plants:*	*Total*	*San Fran*	*Denver*	*Chicago*	*Dallas*	*New York*	
8	S. Carolina	300	0	0	0	80	220	
9	Tennessee	260	0	0	180	80	0	
10	Arizona	280	180	80	20	0	0	
12	Totals:		180	80	200	160	220	
14	*Demands by Whse -->*		180	80	200	160	220	
15	*Plants:*	*Supply*	*Shipping costs from plant x to warehouse y (at intersection):*					
16	S. Carolina	310	10	8	6	5	4	
17	Tennessee	260	6	5	4	3	6	
18	Arizona	280	3	4	5	5	9	
20	*Shipping:*	$3,200	$540	$320	$820	$640	$880	

Formulas relate the various parts of the model.

		Number to ship from plant x to warehouse y (at intersection):				
Plants:	*Total*	*San Fran*	*Denver*	*Chicago*	*Dallas*	*New York*
S. Carolina	=SUM(C8:G8)	0	0	0	80	220
Tennessee	=SUM(C9:G9)	0	0	180	80	0
Arizona	=SUM(C10:G10)	180	80	20	0	0

A separate window contains specifications to Excel's built-in solver to minimize the shipping cost in cell B20, by changing the values of the amounts shipped in cells C8 through G10, while satisfying both demand and capacity constraints.



Instantiation and Replication

No one has the **exact** transportation problem described above. However, this worksheet may be *instantiated* to create a simple application through transformations such as the addition or deletion of rows or columns, and entry of actual data. A click of the solve button then invokes the mathematical optimization routines. This combination of adding information to, and transforming the digital DNA to create an actual application is analogous to the fertilization and incubation of biological DNA. (In the case of the OCR software, instantiation consists of merely placing a printed page into a scanner and clicking an icon on a computer screen.) The degree to which instantiation occurs is directly related to the degree to which the DNA is replicated, which is the primary measure of success in evolving systems.

Of course evolutionary success does not guarantee utility. Just as the DNA of the giant panda is a relative failure compared to that of the cockroach, many potentially useful applications of decision science have failed to catch on. On the other hand, some computer viruses represent successful forms of digital DNA with even lower utility than that of the cockroach.

Classes and Transformations

Geoffrion (1992) in his work on structured modeling, describes *classes* of mathematical models. For example, one can refer abstractly to the class of dense transportation linear programming models such as the one above, without ever mentioning a particular instance of such a problem. There is great expressive power in the ability to describe model classes as opposed to model instances.

The DNA serves as a representative of its class. For it to be an effective modeling device, there must exist simple transformations which map the DNA to any instance of the class. The types of transformations available in a modeling environment strongly influence its ability to map DNA into classes.

Scaling and Hyper-scaling

Two of the most important transformations in this regard are *scaling* and *hyper-scaling*. Scaling changes the cardinality within any dimension of the DNA, as in adding or deleting warehouses or plants in the above transportation model. Hyper-scaling changes the number of dimensions themselves, as in creating a multi-period transportation model by adding a time dimension.

Spreadsheets, Algebraic Modeling Languages, and Multi-dimensional Modelers

Several readily available modeling environments exist in which DNA may be created and transformed.

Spreadsheets

Electronic spreadsheets (such as Microsoft Excel and Lotus 1-2-3) are appealing modeling environments because of their interactivity, and ability to quickly graph results. More importantly, with on the order of 10^7 users, they have become the modeling vernacular among decision makers. This greatly increases the likelihood that spreadsheet DNA will be replicated. However, spreadsheet models are difficult to document, and they scale only moderately, often requiring editing or copying of formulas in the process. They do not hyper-scale beyond 3 dimensions.

Algebraic modeling languages

Modeling languages such as AMPL, GAMS, and LINGO are highly documentable and scalable, and reasonably hyper-scalable. Unfortunately, they are non-interactive in the spreadsheet sense, and because of their algebraic perspective, not easily used directly by decision makers. As a result, they have a user base on the order of only 10^4.

The transportation model DNA as expressed in LINGO appears below.

```
MODEL:

SETS:
WAREHOUSE /SAN_FRAN, DENVER, CHICAGO, DALLAS, NEW_YORK /:DEMAND, RECEIVED;
PLANTS / S_CAROLINA, TENNESSEE, ARIZONA / : CAPACITY, SUPPLIED;
ROUTES(PLANTS, WAREHOUSE)  : VOLUME, COST;
ENDSETS

@FOR(PLANTS(J) : SUPPLIED(J) < CAPACITY(J));
@FOR(WAREHOUSE(I) : RECEIVED(I) > DEMAND(I));
MIN = @SUM( ROUTES(I,J): VOLUME(I,J) * COST(I, J));
@FOR(WAREHOUSE(I) : RECEIVED(I) = @SUM( PLANTS(J): VOLUME(J, I)));
@FOR(PLANTS(J) : SUPPLIED(J) = @SUM( WAREHOUSE(I): VOLUME(J, I)));
DATA:
CAPACITY = 310, 260, 280;
COST = 10, 8, 6, 5, 4, 6, 5, 4, 3, 6, 3, 4, 5, 5, 9;
DEMAND = 180, 80, 200, 160, 220;
ENDDATA

END
```

Multi-dimensional Modelers

A new genre of software products known as multi-dimensional modelers (MDMs) has recently emerged, see PC Magazine (1993) and InfoWorld (1994). Exemplified by Lotus Improv, these are interactive, documentable, and scale and hyper-scale smoothly. They posses some of the best features of both spreadsheets and modeling languages. Because they are new, it is difficult to predict their long term future, but roughly 10^5 have been marketed as of 1994. These packages have the backing of some large software firms and are likely to grow in importance over the next few years.

The transportation model DNA as expressed in IMPROV is shown below.

Model - - TRANS

Sinks

			Warehouses					Total	
			San Fran	Denver	Chicago	Dallas	New York	Shipped	Capacity
Volume	Plants	S Carolina	0	0	0	80	220	300	310
		Tennessee	0	0	180	80	0	260	260
		Arizona	180	80	20	0	0	280	280
	Total Rcvd		180	80	200	160	220		
	Demand		180	80	200	160	220		
Costs	Plants	S Carolina	10	8	6	5	4		
		Tennessee	6	5	4	3	6		
		Arizona	3	4	5	5	9		
	Cost							3200	

Attributes | Sources

✓	1	in Plants:Volume, Total Shipped =sum(Warehouses)
✓	2	in Warehouses:Volume, Total Rcvd =sum(Plants)
✓	3	Costs:Total Shipped:Cost =sumproduct(Volume:Plants:Warehouses ,Costs:Plants:Warehouses)

Although Improv does not provide optimization capability on its own, the author is developing a system for Primal Solutions Inc., under a grant from the Air Force Office of Scientific Research, to aggregate Improv and an algebraic modeling language into a single system. Improv will provide a widely available, interactive platform in which the model will be built and in which the optimization specifications will be written.

Current Optimization Selections - - for - TRANS

✓	1	Objective=Minimize(Model::Costs:Total Shipped:Cost)
✓	2	C1=Constrain(Model::Volume:Warehouses:Total Rcvd , ">" , Model::Warehouses:Volume:Demand)
✓	3	C2=Constrain(Model::Volume:Total Shipped:Plants , "<" , Model::Volume:Plants:Capacity)
✓	4	X1=Positive(Model::Volume:Warehouses:Plants)
	5	

Upon invoking the optimization command, the model along with the optimization specification will be translated into an algebraic modeling language in which it may be solved directly on a PC, or in client server mode on a remote workstation. The LINGO model above was created using this system.

DNA and the Decision Science Curriculum

How should the curriculum of the Research University respond to new stages of technological evolution? Neither too quickly nor too slowly.

Not too quickly, because technologies are changing so fast, that it is impossible for a technical curriculum to track the technological state of the art. Furthermore in its early stages, it is difficult to distinguish a "flash in the pan" from a new technological standard.

Not too slowly, because as technologies become obsolete, or merely sublimated within other technologies, a curriculum becomes irrelevant.

TIME FOR A CHANGE

There is ample evidence that the curricula in the decision science are due for a significant change. In no particular order:

- Enrollment is off, especially in Management Science courses in business schools.
- Decision Science texts and curricula have not changed fundamentally since the microcomputer revolution of 15 years ago.
- Currently the two professional societies in the United States that spearhead the decision sciences, The Operations Research Society of America and The Institute of Management Science, are preparing to aggregate themselves.
- By now it is clear that the microcomputer is not a "flash in pan".

Some Opinionated Suggestions

1. Focus less on algorithms and more on mathematical pitfalls
As algorithms become sublimated within other technologies, such as the linear and nonlinear solvers in millions of copies of Microsoft Excel, it becomes more important to teach concepts related to using these tools rather than on developing new ones. Thus efficient formulations of integer models, and convexity, convergence and stability issues should be stressed. Algorithmists need not despair, however. In the long run, nothing will increase the demand for robust algorithms faster than the current wide dissemination of optimization.

2. Focus less on model formulation and more on model classes and transformations
The direct teaching of mathematical modeling is notoriously difficult and ineffective. It is more effective to introduce the DNA of important model classes and show how it may be transformed to meet individual need. For example, in keeping with my overall theme, once the DNA of the product mix LP and dense transportation LP are presented to students, it is elementary to aggregate these (is this recombinant DNA?) to form more complex models involving both production and shipping.

3. Focus less on Statistics and more on Monte Carlo Simulation
Many people manage to take statistics courses without ever understanding the concept of a probability distribution. Introducing random number generators into interactive mathematical models, and performing Monte Carlo Simulation, can provide an intuitive link to such concepts as the Central Limit Theorem and functions of a random variable.

4. Use Industry Standard Software Where Possible
Students are more likely to benefit from using software that they will see again in the workplace. Do not forget the network externality.

5. Provide DNA on disks

There is no point in taking the DNA approach unless students leave class with a disk containing DNA. See Gardner (1992) and Savage (1993).

6. Teach less to more people

I have found that teaching and providing DNA on disk allows a more rapid coverage of topics. Although all topics may not be absorbed immediately, the student can easily reexamine them experimentally at a later date. The bad news from the decision science faculty perspective is that the total course hours required for a given topic have been reduced. The good news is that there is the potential for a larger audience of students, and the best news of all is that those who go on to become decision makers are more likely than ever to actually use the decision science they are taught.

References

Dawkins, Richard. 1976. *The Selfish Gene*, Oxford University Press, Oxford

Gardner, Everette S. 1992. *The Spreadsheet Operations Manager*, McGraw-Hill, New York.

Geoffrion, Arthur M. 1992. The SML Language for Structured Modeling: Levels 1 and 2. *Operations Research,* **40**, 38-57

InfoWorld, January 24, 1994, *Multidimensional Models Boost Viewing Options,* p.69

PC Magazine, September 28, 1993, *New Dimensions in Spreadsheets,* p.183

Savage, Sam L. 1993. *Fundamental Analytic Spreadsheet Tools for Quantitative Management*, McGraw-Hill, New York

Waldrop, Mitchell M. 1992. *Complexity*, Touchstone, New York